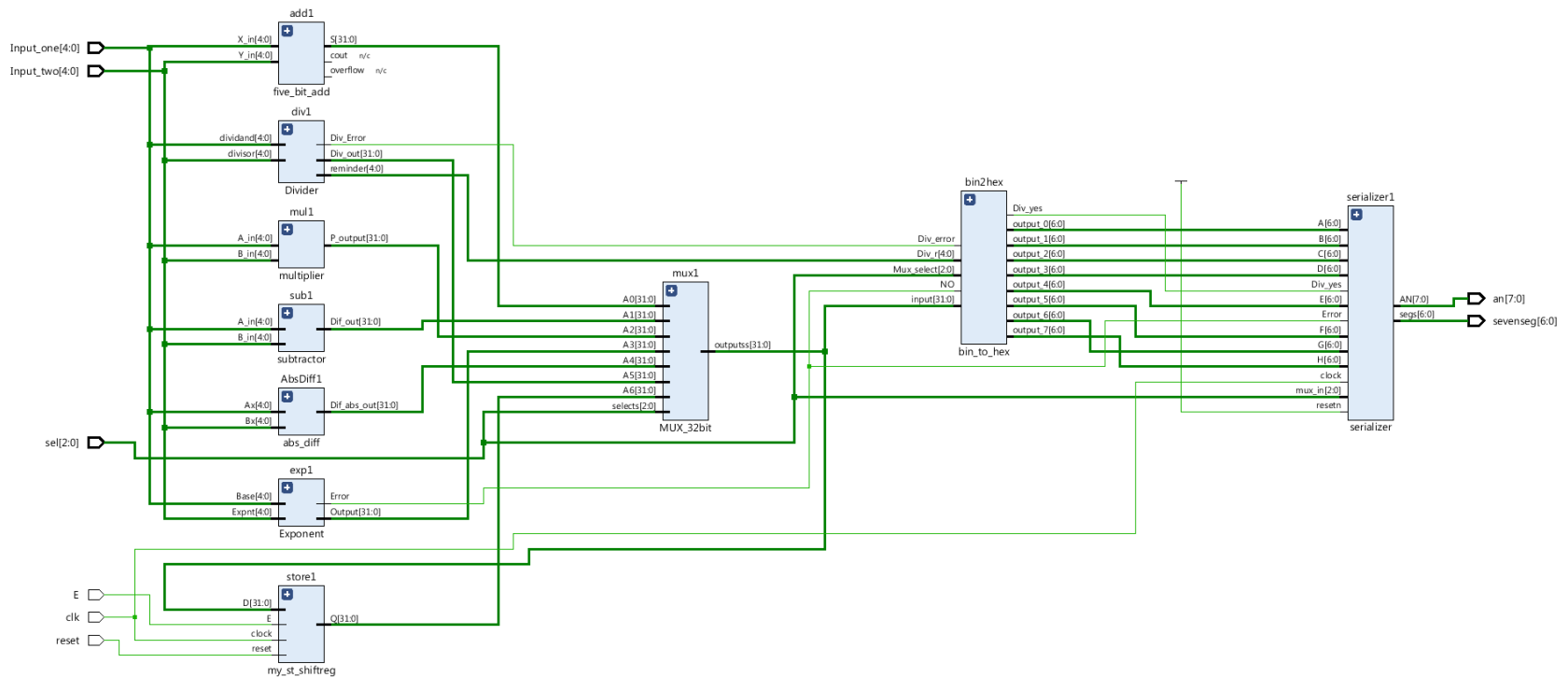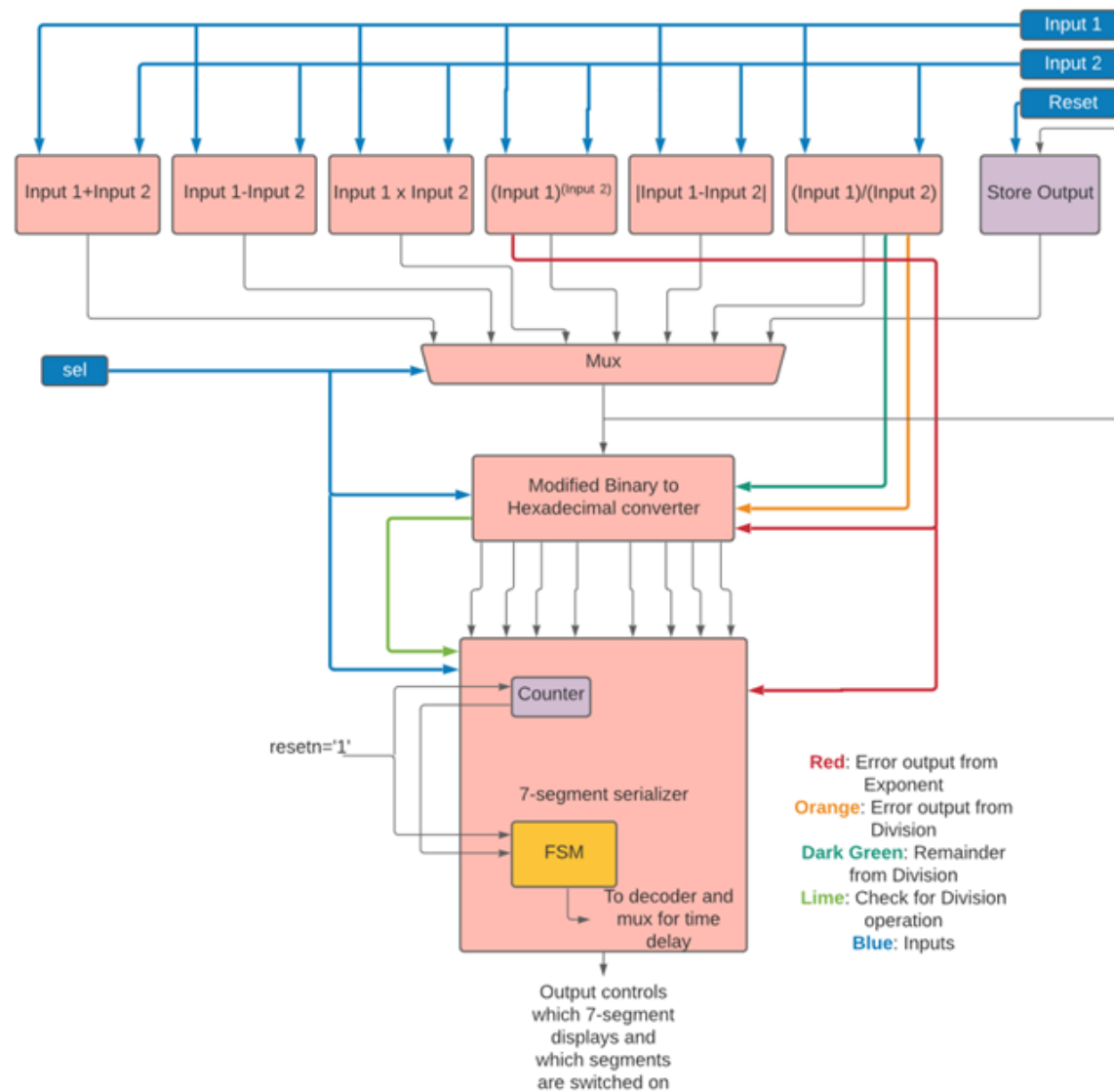# 5-bit signed calculator

Done by: Mohammed Abdul Rafay, Sinan Ghareeb, Mohammed Abdul Wasay

Overall Project Schematic

Control and Datapath

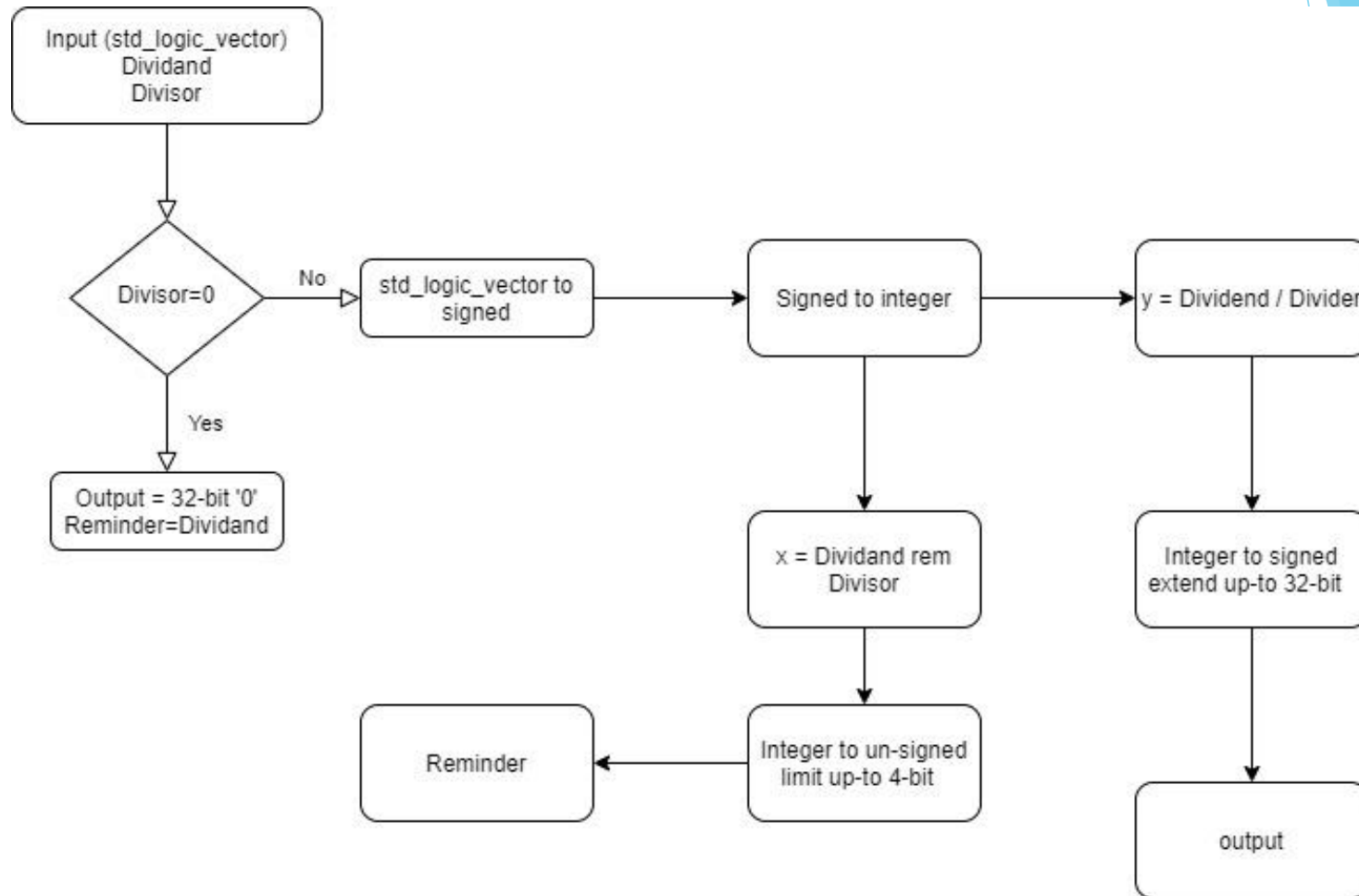# Exponent

- One input is the base
- The other is the exponent
- The inputs are converted into integer and computed
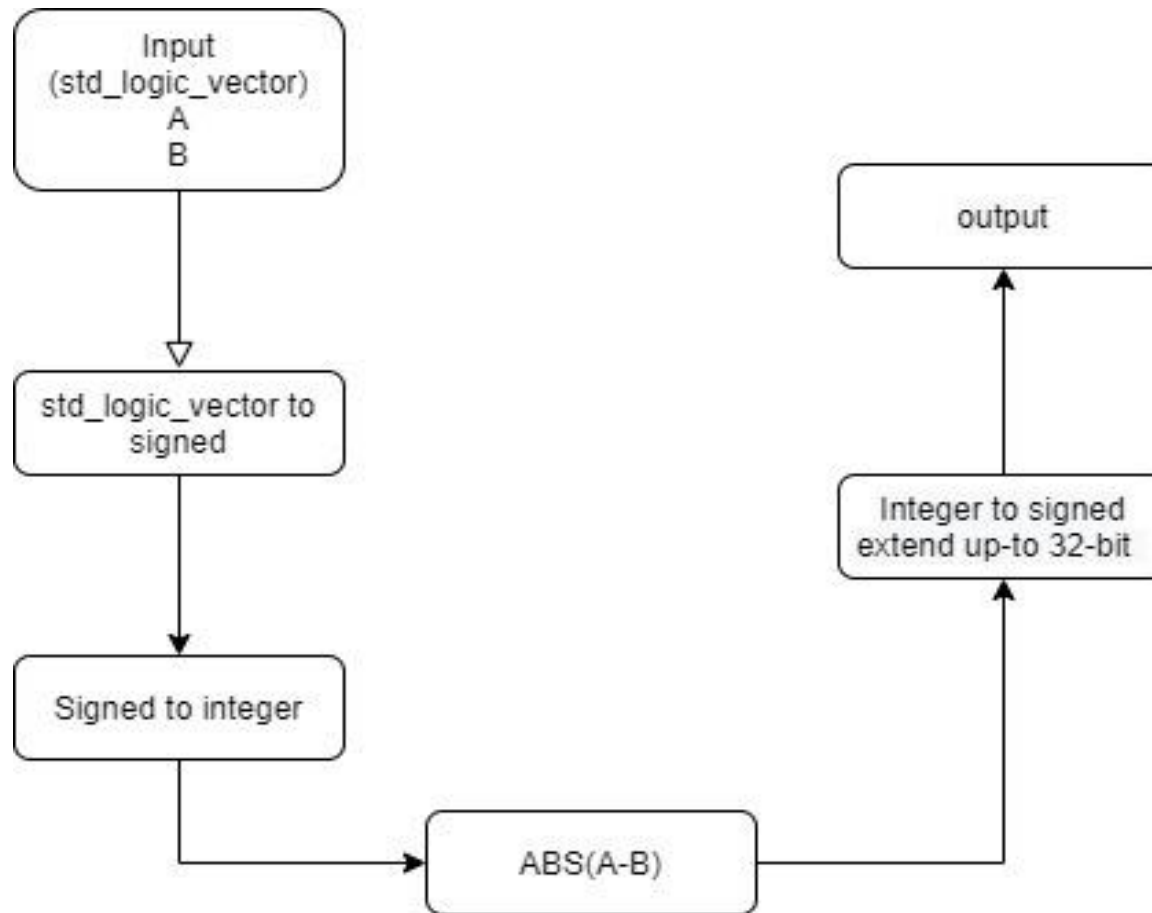- The output is checked using 'if' statements
- There is an Error output

Store function

- Professor Llamocca's parallel access register file was modified.

Signed Divider

Input (std_logic_vector)
Dividand
Divisor

Divisor=0

No → std_logic_vector to signed → Signed to integer → y = Dividend / Divider

Yes

Output = 32-bit '0'
Reminder=Dividand

x = Dividand rem Divisor

Integer to signed extend up-to 32-bit

Reminder ← Integer to un-signed limit up-to 4-bit

output

Absolute Difference

Difference

Signed Addition

Inputs A and B → 7 Full Adders in series (iterative) → Outputs: S, cout, overflow

Signed Multiplication

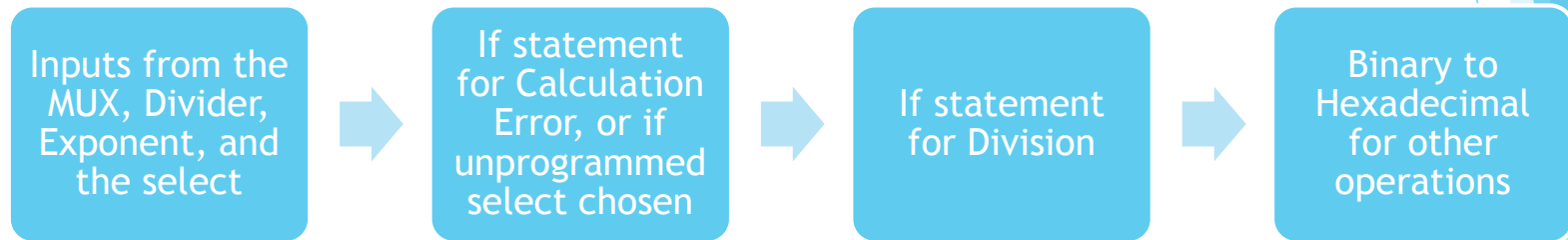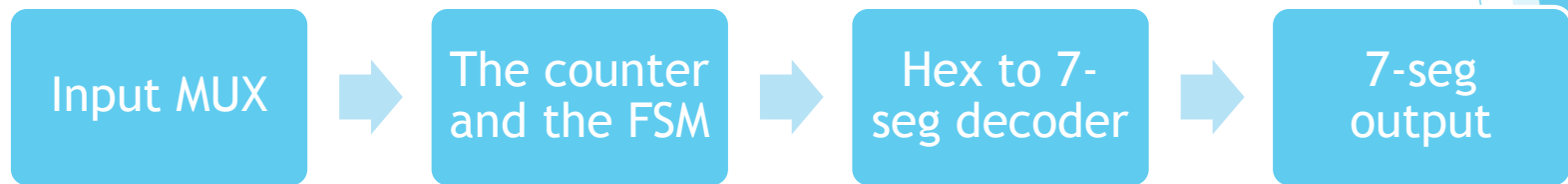| Inputs A and B | → | Sign extension to 31 bits | → | Prof. Llamocca's Array Multiplier | → | Port mapping the result into the output |
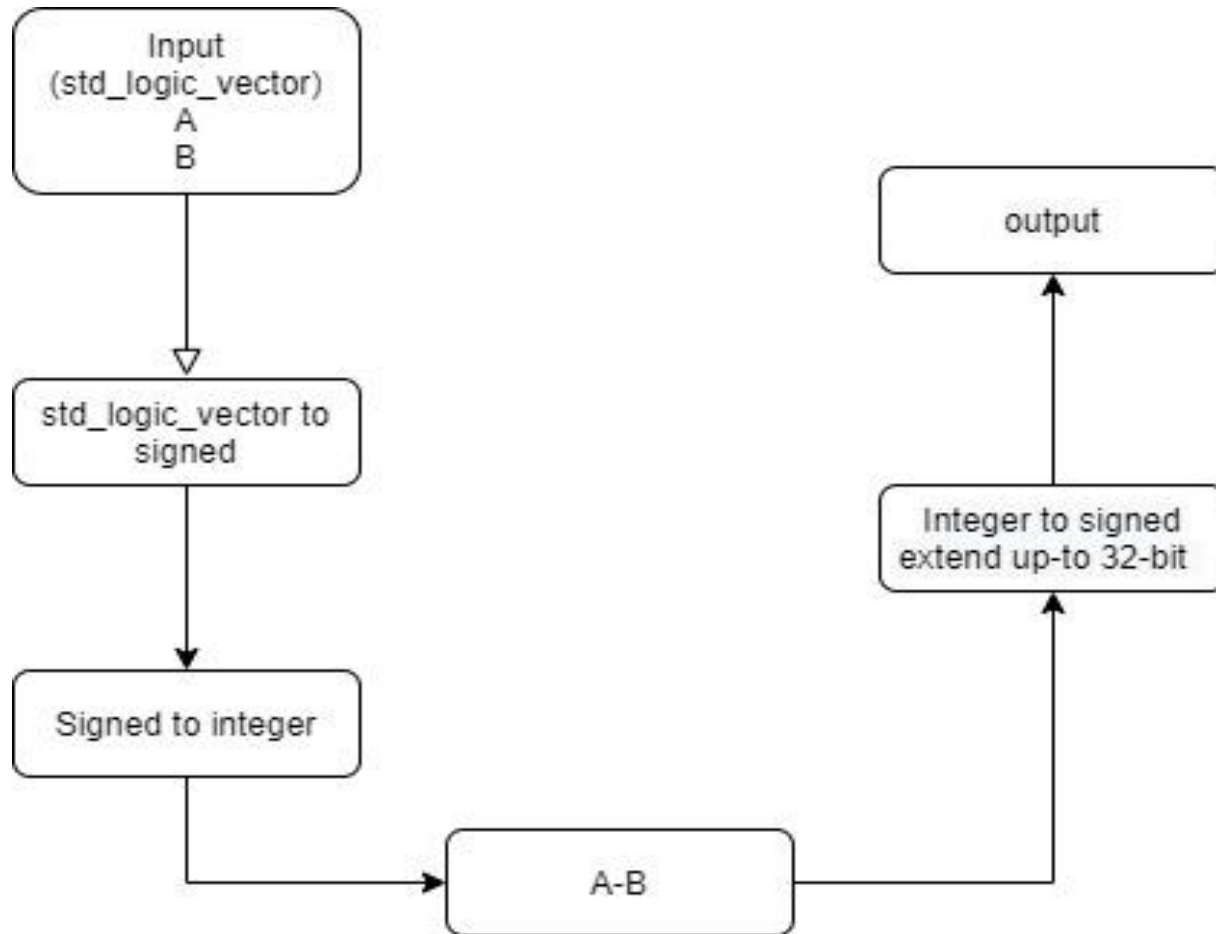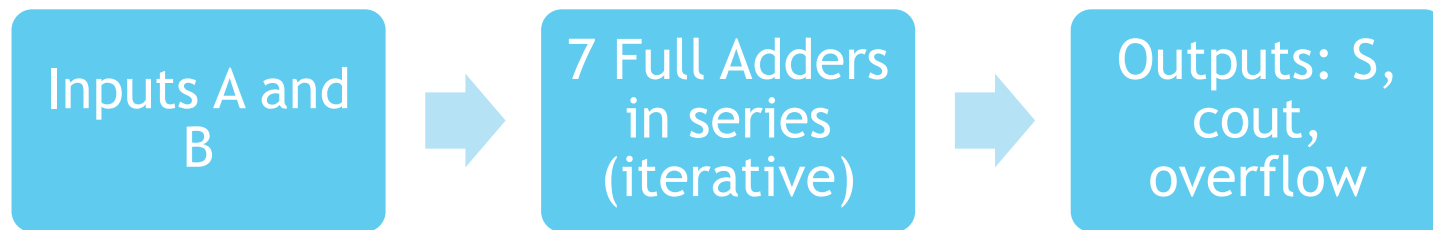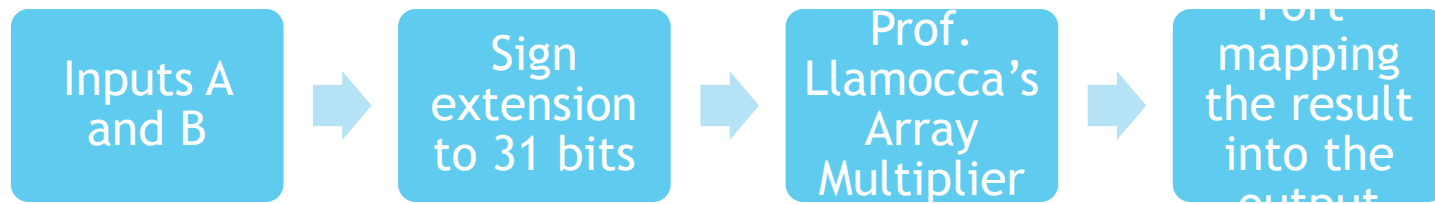
Multiplexor (MUX)

▶ Multiplexers (MUX) are generally used to forward a certain input to the single output at a time.

▶ In ALU MUX's are used to select a certain operator using the opcode(operation code).

▶ In this project 32 and 4-bit design of 8 to 1 Mux are used.

▶ 32-bit 8 to 1 Mux have eight 32-bit busses at the input along with a 3 bit select line input having a single output of 32-bits.

Bin-to-hex converter with special case for Error and Division

| Inputs from the MUX, Divider, Exponent, and the select | → | If statement for Calculation Error, or if unprogrammed select chosen | → | If statement for Division | → | Binary to Hexadecimal for other operations |

7-segment serializer (adapted from Professor Llamocca's serializer)

Input MUX → The counter and the FSM → Hex to 7-seg decoder → 7-seg output

Link to the video

https://youtu.be/WcpUoIuIUdQ

In case the link does not work, the demo video is zipped with this presentation